

UNITED STATES PATENT APPLICATION

FOR

**FILE SYSTEM LEVEL INTEGRITY  
VERIFICATION AND VALIDATION**

INVENTOR:

Christopher Bolin

Prepared by

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN, LLP  
12400 WILSHIRE BOULEVARD  
SEVENTH FLOOR  
LOS ANGELES, CALIFORNIA 90025-1026

(503) 684-6200

Express Mail Label No.: EL034436829US

## **FILE SYSTEM LEVEL INTEGRITY VERIFICATION AND VALIDATION**

### **FIELD OF THE INVENTION**

5           The invention generally relates to scanning data for the presence of certain characteristics, such as virus code, and more particularly to preventing the rescanning of previously scanned files.

### **COPYRIGHT NOTICE/PERMISSION**

10           A portion of the disclosure of this patent document contains material which is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure as it appears in the Patent and Trademark Office patent file or records, but otherwise reserves all copyright rights whatsoever. The following notice applies to the software and data as described below and in the drawings hereto:  
15 Copyright © 2000, Networks Associates Technology, Inc., All Rights Reserved.

### **BACKGROUND**

20           With the advent of general access computer networks, such as the Internet, people may now easily exchange application programs and application data between computer systems.

          Unfortunately, some people have taken advantage of such easy data exchange by developing computer “viruses” designed to spread among and sometimes attack interconnected devices, such as networked computers. A virus is application code that executes on one’s computer without one’s knowledge, and against one’s interests. Viruses tend to replicate themselves within all interconnected devices, allowing an exponential “infection” of other devices.

In response to the security threat intrinsic to viruses, anti-virus programs were developed to identify and remove viruses. Anti-virus programs use virus scanners to scan individual files, groups of files, or the entire hard drive or drives of a computer system for known viruses in a number of ways, such as by comparing each file to a list of "virus signatures" that are stored in "virus signature files" or by emulating computer instructions contained within the file to evaluate the effect of the instructions. The scanning can be done upon request of a user, when the file is accessed on a mass storage device such as by an application, or on a scheduled basis. Virus scanning is, therefore, a resource intensive (both in CPU and disk I/O usage) and time-consuming task, especially in the case of access scanning. Oftentimes, a user's file-open request must be delayed until the file can be scanned and possibly cleaned. This resource consumption can lead to a degradation of a computer's overall performance and slower response times for users.

In light of the foregoing, it is not surprising that a universal problem in anti-virus scanning is the length of time required to scan a file for viruses. This problem has been exacerbated in recent years for a number of reasons. One reason is simply that file sizes are growing ever larger, thereby increasing the average amount of time to scan a given file. Oftentimes, the files are so large that they are compressed and stored as archives. The term archive as used herein includes traditional archive data formats such as ZIP, ZOO, LHA, ARC, JAR, LZW, etc. that contain compressed collections of data files, in addition to other data formats that may embed other files, e.g., Microsoft Word (e.g., ".DOC") documents, Rich Text Format (RTF) files, Object Linking and Embedding (OLE) containers, etc. Scanning archives and documents containing embedded objects takes additional time and resources to scan. In some cases, the virus developers manufacture "malicious" files which purposefully take a long

time to scan, because they themselves are large files such as archives and documents containing embedded objects.

In addition to growing file sizes, another reason why anti-virus scanning technology is taking longer is the growing number of drives in a typical computer system as well as the growth in the size of the drive's storage medium itself. Adding to the problem is the explosive growth in the number of viruses for which the file, groups of files, or drives must be scanned.

One way to reduce the amount of time and computer resources required by anti-virus scanning is to prevent the needless rescanning of a previously scanned file. A number of techniques to prevent rescanning have been implemented in anti-virus scanning technology; however, none of them have adequately addressed the problem. These techniques share the concept of saving a set of parameters, an AV "state," for the file as of the last scan so that once a file has been scanned and found free of infection, it should not need to be scanned again unless the file is modified. The parameters chosen for the AV state are indicative of virus infection if changed, such as the file's length, checksum, and date of last file write operation.

For example, a common approach is to verify the integrity of a file by using a cyclical redundancy check (CRC) which generates a unique number when applied to a given file, often referred to as a checksum. The checksum remains unchanged as long as the file is unchanged. However, generating and verifying the checksum for each file is time-consuming and often uses more resources than simply performing the anti-virus scan itself. Indeed, the cost of generating a checksum for every file on every scan prohibits implementation given the constraints on the processing power and speed of today's computer systems.

Another approach is to maintain a database of file date, time, and size information, and compare the files to be scanned against the database to determine if there is a difference which

indicates that the file should be scanned. This approach has proved easy to circumvent, however, by redirecting the database updates to another file.

What is needed, therefore, is an improved method of preventing the needless rescanning of previously scanned files. The above-mentioned shortcomings, disadvantages and problems  
5 are addressed by the present invention, which will be understood by reading and studying the following specification.

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
1000  
1001  
1002  
1003  
1004  
1005  
1006  
1007  
1008  
1009  
1010  
1011  
1012  
1013  
1014  
1015  
1016  
1017  
1018  
1019  
1020  
1021  
1022  
1023  
1024  
1025  
1026  
1027  
1028  
1029  
1030  
1031  
1032  
1033  
1034  
1035  
1036  
1037  
1038  
1039  
1040  
1041  
1042  
1043  
1044  
1045  
1046  
1047  
1048  
1049  
1050  
1051  
1052  
1053  
1054  
1055  
1056  
1057  
1058  
1059  
1060  
1061  
1062  
1063  
1064  
1065  
1066  
1067  
1068  
1069  
1070  
1071  
1072  
1073  
1074  
1075  
1076  
1077  
1078  
1079  
1080  
1081  
1082  
1083  
1084  
1085  
1086  
1087  
1088  
1089  
1090  
1091  
1092  
1093  
1094  
1095  
1096  
1097  
1098  
1099  
1100  
1101  
1102  
1103  
1104  
1105  
1106  
1107  
1108  
1109  
1110  
1111  
1112  
1113  
1114  
1115  
1116  
1117  
1118  
1119  
1120  
1121  
1122  
1123  
1124  
1125  
1126  
1127  
1128  
1129  
1130  
1131  
1132  
1133  
1134  
1135  
1136  
1137  
1138  
1139  
1140  
1141  
1142  
1143  
1144  
1145  
1146  
1147  
1148  
1149  
1150  
1151  
1152  
1153  
1154  
1155  
1156  
1157  
1158  
1159  
1160  
1161  
1162  
1163  
1164  
1165  
1166  
1167  
1168  
1169  
1170  
1171  
1172  
1173  
1174  
1175  
1176  
1177  
1178  
1179  
1180  
1181  
1182  
1183  
1184  
1185  
1186  
1187  
1188  
1189  
1190  
1191  
1192  
1193  
1194  
1195  
1196  
1197  
1198  
1199  
1200  
1201  
1202  
1203  
1204  
1205  
1206  
1207  
1208  
1209  
1210  
1211  
1212  
1213  
1214  
1215  
1216  
1217  
1218  
1219  
1220  
1221  
1222  
1223  
1224  
1225  
1226  
1227  
1228  
1229  
1230  
1231  
1232  
1233  
1234  
1235  
1236  
1237  
1238  
1239  
1240  
1241  
1242  
1243  
1244  
1245  
1246  
1247  
1248  
1249  
1250  
1251  
1252  
1253  
1254  
1255  
1256  
1257  
1258  
1259  
1260  
1261  
1262  
1263  
1264  
1265  
1266  
1267  
1268  
1269  
1270  
1271  
1272  
1273  
1274  
1275  
1276  
1277  
1278  
1279  
1280  
1281  
1282  
1283  
1284  
1285  
1286  
1287  
1288  
1289  
1290  
1291  
1292  
1293  
1294  
1295  
1296  
1297  
1298  
1299  
1300  
1301  
1302  
1303  
1304  
1305  
1306  
1307  
1308  
1309  
1310  
1311  
1312  
1313  
1314  
1315  
1316  
1317  
1318  
1319  
1320  
1321  
1322  
1323  
1324  
1325  
1326  
1327  
1328  
1329  
1330  
1331  
1332  
1333  
1334  
1335  
1336  
1337  
1338  
1339  
1340  
1341  
1342  
1343  
1344  
1345  
1346  
1347  
1348  
1349  
1350  
1351  
1352  
1353  
1354  
1355  
1356  
1357  
1358  
1359  
1360  
1361  
1362  
1363  
1364  
1365  
1366  
1367  
1368  
1369  
1370  
1371  
1372  
1373  
1374  
1375  
1376  
1377  
1378  
1379  
1380  
1381  
1382  
1383  
1384  
1385  
1386  
1387  
1388  
1389  
1390  
1391  
1392  
1393  
1394  
1395  
1396  
1397  
1398  
1399  
1400  
1401  
1402  
1403  
1404  
1405  
1406  
1407  
1408  
1409  
1410  
1411  
1412  
1413  
1414  
1415  
1416  
1417  
1418  
1419  
1420  
1421  
1422  
1423  
1424  
1425  
1426  
1427  
1428  
1429  
1430  
1431  
1432  
1433  
1434  
1435  
1436  
1437  
1438  
1439  
1440  
1441  
1442  
1443  
1444  
1445  
1446  
1447  
1448  
1449  
1450  
1451  
1452  
1453  
1454  
1455  
1456  
1457  
1458  
1459  
1460  
1461  
1462  
1463  
1464  
1465  
1466  
1467  
1468  
1469  
1470  
1471  
1472  
1473  
1474  
1475  
1476  
1477  
1478  
1479  
1480  
1481  
1482  
1483  
1484  
1485  
1486  
1487  
1488  
1489  
1490  
1491  
1492  
1493  
1494  
1495  
1496  
1497  
1498  
1499  
1500  
1501  
1502  
1503  
1504  
1505  
1506  
1507  
1508  
1509  
1510  
1511  
1512  
1513  
1514  
1515  
1516  
1517  
1518  
1519  
1520  
1521  
1522  
1523  
1524  
1525  
1526  
1527  
1528  
1529  
1530  
1531  
1532  
1533  
1534  
1535  
1536  
1537  
1538  
1539  
1540  
1541  
1542  
1543  
1544  
1545  
1546  
1547  
1548  
1549  
1550  
1551  
1552  
1553  
1554  
1555  
1556  
1557  
1558  
1559  
1560  
1561  
1562  
1563  
1564  
1565  
1566  
1567  
1568  
1569  
1570  
1571  
1572  
1573  
1574  
1575  
1576  
1577  
1578  
1579  
1580  
1581  
1582  
1583  
1584  
1585  
1586  
1587  
1588  
1589  
1590  
1591  
1592  
1593  
1594  
1595  
1596  
1597  
1598  
1599  
1600  
1601  
1602  
1603  
1604  
1605  
1606  
1607  
1608  
1609  
1610  
1611  
1612  
1613  
1614  
1615  
1616  
1617  
1618  
1619  
1620  
1621  
1622  
1623  
1624  
1625  
1626  
1627  
1628  
1629  
1630  
1631  
1632  
1633  
1634  
1635  
1636  
1637  
1638  
1639  
1640  
1641  
1642  
1643  
1644  
1645  
1646  
1647  
1648  
1649  
1650  
1651  
1652  
1653  
1654  
1655  
1656  
1657  
1658  
1659  
1660  
1661  
1662  
1663  
1664  
1665  
1666  
1667  
1668  
1669  
1670  
1671  
1672  
1673  
1674  
1675  
1676  
1677  
1678  
1679  
1680  
1681  
1682  
1683  
1684  
1685  
1686  
1687  
1688  
1689  
1690  
1691  
1692  
1693  
1694  
1695  
1696  
1697  
1698  
1699  
1700  
1701  
1702  
1703  
1704  
1705  
1706  
1707  
1708  
1709  
1710  
1711  
1712  
1713  
1714  
1715  
1716  
1717  
1718  
1719  
1720  
1721  
1722  
1723  
1724  
1725  
1726  
1727  
1728  
1729  
1730  
1731  
1732  
1733  
1734  
1735  
1736  
1737  
1738  
1739  
1740  
1741  
1742  
1743  
1744  
1745  
1746  
1747  
1748  
1749  
1750  
1751  
1752  
1753  
1754  
1755  
1756  
1757  
1758  
1759  
1760  
1761  
1762  
1763  
1764  
1765  
1766  
1767  
1768  
1769  
1770  
1771  
1772  
1773  
1774  
1775  
1776  
1777  
1778  
1779  
1780  
1781  
1782  
1783  
1784  
1785  
1786  
1787  
1788  
1789  
1790  
1791  
1792  
1793  
1794  
1795  
1796  
1797  
1798  
1799  
1800  
1801  
1802  
1803  
1804  
1805  
1806  
1807  
1808  
1809  
1810  
1811  
1812  
1813  
1814  
1815  
1816  
1817  
1818  
1819  
1820  
1821  
1822  
1823  
1824  
1825  
1826  
1827  
1828  
1829  
1830  
1831  
1832  
1833  
1834  
1835  
1836  
1837  
1838  
1839  
1840  
1841  
1842  
1843  
1844  
1845  
1846  
1847  
1848  
1849  
1850  
1851  
1852  
1853  
1854  
1855  
1856  
1857  
1858  
1859  
1860  
1861  
1862  
1863  
1864  
1865  
1866  
1867  
1868  
1869  
1870  
1871  
1872  
1873  
1874  
1875  
1876  
1877  
1878  
1879  
1880  
1881  
1882  
1883  
1884  
1885  
1886  
1887  
1888  
1889  
1890  
1891  
1892  
1893  
1894  
1895  
1896  
1897  
1898  
1899  
1900  
1901  
1902  
1903  
1904  
1905  
1906  
1907  
1908  
1909  
1910  
1911  
1912  
1913  
1914  
1915  
1916  
1917  
1918  
1919  
1920  
1921  
1922  
1923  
1924  
1925  
1926  
1927  
1928  
1929  
1930  
1931  
1932  
1933  
1934  
1935  
1936  
1937  
1938  
1939  
1940  
1941  
1942  
1943  
1944  
1945  
1946  
1947  
1948  
1949  
1950  
1951  
1952  
1953  
1954  
1955  
1956  
1957  
1958  
1959  
1960  
1961  
1962  
1963  
1964  
1965  
1966  
1967  
1968  
1969  
1970  
1971  
1972  
1973  
1974  
1975  
1976  
1977  
1978  
1979  
1980  
1981  
1982  
1983  
1984  
1985  
1986  
1987  
1988  
1989  
1990  
1991  
1992  
1993  
1994  
1995  
1996  
1997  
1998  
1999  
2000  
2001  
2002  
2003  
2004  
2005  
2006  
2007  
2008  
2009  
2010  
2011  
2012  
2013  
2014  
2015  
2016  
2017  
2018  
2019  
2020  
2021  
2022  
2023  
2024  
2025  
2026  
2027  
2028  
2029  
2030  
2031  
2032  
2033  
2034  
2035  
2036  
2037  
2038  
2039  
2040  
2041  
2042  
2043  
2044  
2045  
2046  
2047  
2048  
2049  
2050  
2051  
2052  
2053  
2054  
2055  
2056  
2057  
2058  
2059  
2060  
2061  
2062  
2063  
2064  
2065  
2066  
2067  
2068  
2069  
2070  
2071  
2072  
2073  
2074  
2075  
2076  
2077  
2078  
2079  
2080  
2081  
2082  
2083  
2084  
2085  
2086  
2087  
2088  
2089  
2090  
2091  
2092  
2093  
2094  
2095  
2096  
2097  
2098  
2099  
2100  
2101  
2102  
2103  
2104  
2105  
2106  
2107  
2108  
2109  
2110  
2111  
2112  
2113  
2114  
2115  
2116  
2117  
2118  
2119  
2120  
2121  
2122  
2123  
2124  
2125  
2126  
2127  
2128  
2129  
2130  
2131  
2132  
2133  
2134  
2135  
2136  
2137  
2138  
2139  
2140  
2141  
2142  
2143  
2144  
2145  
2146  
2147  
2148  
2149  
2150  
2151  
2152  
2153  
2154  
2155  
2156  
2157  
2158  
2159  
2160  
2161  
2162  
2163  
2164  
2165  
2166  
2167  
2168  
2169  
2170  
2171  
2172  
2173  
2174  
2175  
2176  
2177  
2178  
2179  
2180  
2181  
2182  
2183  
2184  
2185  
2186  
2187  
2188  
2189  
2190  
2191  
2192  
2193  
2194  
2195  
2196  
2197  
2198  
2199  
2200  
2201  
2202  
2203  
2204  
2205  
2206  
2207  
2208  
2209  
2210  
2211  
2212

## SUMMARY OF THE INVENTION

According to one aspect of the present invention, an integrity verifier and validator performs an initial anti-virus scan against an entire file system to verify the integrity of the file system and create a corresponding file system level integrity verification database of known scanned regions of the file system. The database contains a copy of the file system's partition table referencing the range of occupied inodes and directory blocks. When new or updated content is written to the file system the integrity of the corresponding occupied inodes and directory blocks is no longer assured, and the integrity verifier flags them on the database for rescanning.

According to one aspect of the present invention, subsequent attempts to rescan any portion of the file system triggers the integrity verifier and validator to scan the database to validate whether that portion of the file system falls within any of the occupied inodes or directory blocks that have been flagged for rescanning. If they are not flagged, then rescanning of that portion of the file system is unnecessary as the integrity of at least that portion is assured. However, if they are flagged, then the anti-virus software may rescan that portion of the file system for the presence of viruses.

Because the integrity scan is performed against the integrity verification database at the higher file system level instead of the lower individual file or directory level, the processing overhead is less, resulting in a more efficient method of preventing needless rescanning of previously scanned files. While it is contemplated that the integrity verifier and validator will be used for requests to scan entire hard drives, disks, CD-ROMs and the like, it could just as easily be used for requests to scan individual objects such as individual files or archives. However, the larger the portion of the file system that is requested to be rescanned, the greater the level of

efficiency that can be achieved using the present invention to prevent needless rescanning. In addition to the aspects and advantages of the present invention described in this summary, further aspects and advantages of the invention will become apparent by reference to the drawings and by reading the detailed description that follows.

Patented by the U.S. Patent and Trademark Office

## BRIEF DESCRIPTION OF THE DRAWINGS

Features and advantages of the invention will become apparent to one skilled in the art to which the invention pertains from a review of the following detailed description and claimed  
5 embodiments of the invention in conjunction with the drawings, and in which:

FIG. 1 is a block diagram illustrating one generalized embodiment of a file system level integrity verification and validation process incorporating the invention, and the operating environment in which certain aspects of the illustrated invention may be practiced;

FIG. 2 is a flow diagram illustrating certain aspects of a method to be performed by a  
10 computer executing one embodiment of the illustrated invention shown in FIG. 1;

FIG. 3 is a flow diagram illustrating certain aspects of a method to be performed by a computer executing one embodiment of the illustrated invention shown in FIG. 1;

FIG. 4 illustrates a suitable computing environment in which certain aspects of the illustrated invention shown in FIGS. 1-3 may be practiced.

## DETAILED DESCRIPTION

In the following description various aspects of the present invention, a method and apparatus for verifying and validating the integrity of an object, will be described. Specific details will be set forth in order to provide a thorough understanding of the present invention. However, it will be apparent to those skilled in the art that the present invention may be practiced  
20 with only some or all of the described aspects of the present invention, and with or without some or all of the specific details. In some instances, well-known features may be omitted or simplified in order not to obscure the present invention.



Parts of the description will be presented using terminology commonly employed by those skilled in the art to convey the substance of their work to others skilled in the art, including terms of operations performed by a computer system or virus scanner, and their operands. As well understood by those skilled in the art, these operands take the form of electrical, magnetic, or optical signals, and the operations involve storing, transferring, combining, and otherwise manipulating the signals through electrical, magnetic or optical components of a system. The term system includes general purpose as well as special purpose arrangements of these components that are standalone, adjunct or embedded.

Various operations will be described as multiple discrete steps performed in turn in a manner that is most helpful in understanding the present invention. However, the order of description should not be construed as to imply that these operations are necessarily performed in the order they are presented, or even order dependent. Lastly, repeated usage of the phrase “in one embodiment” does not necessarily refer to the same embodiment, although it may.

A virus scanner must be capable of quickly detecting and disposing of malicious files, emails, archives, or other objects that may be infected with a virus. Otherwise, accessing the malicious object can result in the temporary or even prolonged disablement or stalling of all virus scanning and/or file activity of the scanner’s host system. When scanning large collections of objects, such as entire hard drives, disks, or CD-ROMs, any delay associated with scanning is compounded. When the virus scanner is disabled, the user is vulnerable to subsequent viral attacks. In some host systems, the virus scanner may be integrated into the operating system in such a manner that the disablement of the virus scanner causes the entire system to become unresponsive or appear to hang.

One of the ways in which to reduce the amount of time required to perform the virus scanner is to prevent needless rescanning of objects that have previously been scanned. The method and apparatus of the present invention prevents needless rescanning of objects that have previously been scanned by validating the integrity of the object at the file system level. A file system refers to the data files (i.e. objects) in a computer system and the file management system used to keep track of those objects. The validation of the integrity of objects at the file system level is performed by taking advantage of information present in the logical structures of the storage medium in which the objects of the file system are stored. The storage medium may be a hard disk, a disk drive, a CD-ROM , and the like, all of which are referred to herein as a storage medium.

The information present in the storage medium's logical structures is maintained by the operating system's file management and real-time monitoring systems. The storage medium's logical structures may include a partition table, which defines how many partitions are formatted on the storage medium. A partition of a storage medium is a logical division of the medium created so that different operating systems can co-exist on the same medium, or to create the appearance of having separate storage media for file management, multiple users, or other purposes. The file management system maintains certain data in the partition table, such as a super-block, which contains information about the file system as a whole, an inode, which contains all file information except the file name, a directory block, which contains the file name and the number of the inode that represents the file, and a data block, which contains the location in the file system where the actual file data is stored. Taken together, the partition table data indicate the identity, location, and status (i.e. changed or not changed) of an object or group of objects in the file system.

It should be noted that the terms partition table, super-block, inode, directory block, and data block, are used herein as generic terms that describe those logical structures which are maintained by a typical file management system, including such file management systems as the File Allocation Table (FAT) system supported by most operating systems as Microsoft Windows NT, Windows 98, the Apple Macintosh OS, and some versions of UNIX operating system. Besides FAT, other file management systems include the Virtual File Allocation Table (VFAT), which handles long file names, or the File Allocation Table 32 (FAT32), which, among other advantages, allows for larger partition sizes. Yet another type of file management system is the New Technology File System (NTFS), which allows for increased fault tolerance and enhanced security as compared to other file systems. The intricacies of these and other file management systems are well-known to those skilled in the art and thus will not be discussed further except as they pertain to the present invention. It should be noted, however, that any of the above-mentioned and other file management systems may be employed without departing from the scope of the present invention.

The real-time monitoring system monitors file system events such as opening or changing an object, and creates entries in the file system partition table to record such events. The entries identify which of the super-blocks, inodes, directory blocks, and data blocks of the storage medium are occupied by the opened or changed object or objects. As with the file management systems, a number of different real-time monitoring systems may be employed without departing from the scope of the present invention. For example, the real-time monitoring system may be a VxD device driver, a network loadable module, or a UNIX daemon. The intricacies of these and other real-time monitoring systems are well-known to those skilled in the art and thus will not be discussed further except as they pertain to the present invention. It should be noted, however,

that any of the above-mentioned and other real-time monitoring systems may be employed without departing from the scope of the present invention.

FIG. 1 is a block diagram illustrating one generalized embodiment of a file system level integrity verification and validation process 100 incorporating the invention, and the operating environment in which certain aspects of the illustrated invention may be practiced. The illustrated file system level integrity verification and validation process 100 operates in conjunction with a real time monitoring system 200, a file system 300 and an AV scanner 400. An example of an AV scanner 400 that may be employed in the present invention is the anti-virus program sold under the registered trademark, McAfee VirusScan, by Network Associates, Inc., of Santa Clara, California. However, other AV scanners may be employed without departing from the scope of the present invention.

The real-time monitoring system 200 is known in the art, and monitors file system events such as an attempt by an application 210 to open or modify a data file 220. Each file system event causes an entry to be made in a file system partition table 230 in accordance with the operating system's file system 300. The file system 300 is also known in the art, and is comprised of, among other things, the data files 220 and an associated file system partition table 230. The file system partition table 230 is comprised of a number of data structures 240 that keep track of the data files 220 on a disk or a partition of a storage medium, and the way the data files 220 are organized.

The real-time monitoring system 200 and file system 300, including the file system partition table 230, and associated data structures 240, may vary from one operating system to another. But all operating systems have some type of real-time operating system 200 and file system 300 that create and maintain a partition table-like collection of data structures that are

identical or similar to those illustrated in the partition table 230 and corresponding data structures 240. For example, the illustrated file system level integrity verification and validation process 100 may be performed on Apple Macintosh, UNIX, or Microsoft file objects, as long as the underlying operating system has some type of real-time monitoring and file systems to manage the objects within that system. In addition, the illustrated file system level integrity verification and validation process 100 may be performed on container objects or embedded objects using the real-time monitoring and file systems that manage those objects, such as Microsoft's Object Linking and Embedding (OLE) technology, or the Component Object Model (COM) technology.

In one embodiment, data structures 240 include a super block 241, which contains information about the file system 300 as a whole, an inode 242, which contains all file information except the file name, the directory block 243, which contains the file name and the number of the inode that represents the file, and the data block 244, which contains the location in the file system 300 where the actual file data is stored. Other combinations of data structures 240 may be employed without departing from the scope of the present invention.

In one embodiment, an integrity verifier and validator 110 performs an initial anti-virus scan against an entire file system to verify the integrity of the file system and create a corresponding file system level integrity verification database 130. The database 130 contains a copy of the file system's partition table data structures 240, including entries for the corresponding super blocks 141, inodes 142, directory blocks 143, and data blocks 144. In one embodiment, the database 130 may contain a copy of some or all of the data structures 240 without departing from the scope of the invention. In particular, the database 130 may include entries referencing the range of occupied inodes 142 and directory blocks 143. An occupied inode 142 and directory block 143 indicates the presence of a file system object such as a file or

a directory (i.e. a collection of files) in the file system 300. When new or updated content is detected by the real time monitoring system 200 causing updates to the file system partition table 230 and corresponding entries 240, then the integrity of the data in the corresponding occupied inodes 141 and directory blocks 142 is no longer assured, and the integrity verifier 110 flags the inodes 142 and directory blocks 143 on the database 130 for rescanning.

The flagging for rescanning may be accomplished in a number of ways, such as by setting a data field on the database 130 to an "on" or "off" value, where the "on" value is the decimal or binary value of one, and the "off" value is the decimal or binary value of zero. Other types of flagging may be used without departing from the principles of the present invention.

Subsequent attempts to rescan any portion of the file system 300 monitored by the real-time monitoring system 200, triggers the integrity validator 110 to validate whether that portion of the file system falls within any of the occupied inodes 141 or directory blocks 142 that have been flagged for rescanning. If they are not flagged, then rescanning of that portion of the file system is unnecessary as the integrity of at least that portion is assured. However, if they are flagged, then the AV Scanner 300 may rescan that portion of the file system for the presence of viruses.

Turning now to FIGS. 2-3, the particular methods of the invention are described in terms of computer software with reference to a series of flowcharts. The methods to be performed by a computer constitute computer programs made up of computer-executable instructions.

Describing the methods by reference to a flowchart enables one skilled in the art to develop such programs including such instructions to carry out the methods on suitably configured computers (the processor of the computer executing the instructions from computer-accessible media). The computer-executable instructions may be written in a computer programming language or may

be embodied in firmware logic. If written in a programming language conforming to a recognized standard, such instructions can be executed on a variety of hardware platforms and for interface to a variety of operating systems. In addition, the present invention is not described with reference to any particular programming language. It will be appreciated that a variety of programming languages may be used to implement the teachings of the invention as described herein. Furthermore, it is common in the art to speak of software, in one form or another (e.g., program, procedure, process, application...), as taking an action or causing a result. Such expressions are merely a shorthand way of saying that execution of the software by a computer causes the processor of the computer to perform an action or a produce a result.

Referring to FIG. 2, certain aspects of the file system level integrity verification and validation process 100 to verify the integrity of the file system are shown. In particular, FIG. 2 illustrates some of the acts to be performed by a computer executing an integrity verifier and validator 110 that incorporates one embodiment of the invention.

In one embodiment, at processing block 112, the integrity verifier and validator 110 prepares for validating the integrity of objects (described below beginning with block 120 in FIG. 3) by scanning the entire file system 300 using the AV Scanner 400. At block 114, by scanning the entire file system 300, the integrity verifier 110 is able to create an integrity verification database 130 of known scanned regions of the file system 300. A known scanned region may comprise one or more objects that do not need to be rescanned until they are changed by some file system event subsequent to the scan of the entire file system at block 114. The known scanned regions are reflected in the integrity verification database 130 by one or more super-, directory, and data blocks 141, 143, and 144 and inodes 142 that are initially unflagged, i.e. objects occupying those data blocks 141, 143, and 144 and inodes 142 do not need to be

rescanned until such time as the data blocks 141, 143, and 144 and inodes 142 are flagged for rescanning as described at block 116/118 below.

At block 116, the integrity verifier 110 receives real-time monitoring events from real-time monitoring system 200. The real-time monitoring events may indicate that an object has been accessed in some way, including reading, opening, creating, or changing the object. In one embodiment, the real-time monitoring events from real-time monitoring system 200 take the form of information from a VxD device driver program running on the Microsoft Windows operating system. The VxD program performs the actual real-time monitoring of file system events. In an alternate embodiment, the real-time monitoring events from real-time monitoring system 200 take the form of information provided by a Netware Network Loadable Module (NLM), or a UNIX daemon. In one embodiment, at block 118, the integrity verifier and validator device 110 flags the super-, directory, and data blocks 141, 143, and 144 and inodes 142 in the integrity verification database 130 that correspond to the super-, directory, and data blocks 241, 243, and 244 and inodes 242 in the file system partition table 230 for which a real-time monitoring event has been detected. The integrity verifier and validator 110 processes described in blocks 116 and 118 are repeated continuously as long as the real-time monitoring system 200 is active.

Referring now to FIG. 3, certain aspects of the file system level integrity verification and validation process 100 to validate the integrity of a file system object are shown. In particular, FIG. 3 illustrates some of the acts to be performed by a computer executing an integrity verifier and validator 110 that incorporates one embodiment of the invention.

In one embodiment, at processing block 120, the integrity validator 110 begins validating the integrity of an object by receiving an integrity verification request from an AV Scanner 400.



In one embodiment, the integrity verifier and validator 110 bypasses the actual integrity validation process if, at block 121, it determines that the AV Signature file 410 has changed since the last time that the object of the integrity verification request was scanned. The AV Signature file contains the latest signatures of known viruses for which the AV Scanner 400 should scan. If the AV Signature file 410 has changed, then the integrity verifier and validator 110 branches directly to performing the AV Scanner 300 at block 124. Otherwise, the integrity verifier and validator 110 continues at block 122 to obtain the block and inode identifiers for the object of the integrity verification request. The block and inode identifiers indicate which of the blocks 141/143/144 and inodes 142 of file system 300 are occupied by the object of the integrity verification request. When the object of the integrity verification request is simply a single data file 220, then there is likely only one occupied block 141/143/144 and inode 142 that need to be validated. When, as is more typically the case, the object of the integrity verification request is a collection of objects, such as an entire floppy disk, or an entire directory of data files 220, then there are likely several occupied blocks 141/143/144 and inodes 142 that need to be validated.

In one embodiment, at block 123, integrity verifier and validator 110 validates the object of the integrity verification request by determining whether the identified occupied blocks 141/143/144 and inodes 142 have been flagged during the verification process described at blocks 112 - 118 of the integrity verifier and validator 110. If the identified occupied blocks 141/143/144 and inodes 142 are flagged for rescanning, then at block 124, the integrity verifier and validator 110 performs the AV Scanner 300 against the object of the integrity verification request. After performing the AV Scanner 300, the integrity verifier and validator 110 updates the integrity verification database 130 to remove the flags from the identified occupied blocks 141/143/144 and inodes 142. By removing the flags, the integrity verifier and validator 110

insures that the next time an object of the integrity verification request which identifies those occupied blocks 141/143/144 and inodes 142, that the AV Scanner will be bypassed, as rescanning is unnecessary as long as the integrity of the object is validated. Finally, at block 126, the object of the integrity verification request is returned to the requester.

5           However, if the identified occupied blocks 141/143/144 and inodes 142 are not flagged, then the integrity verifier and validator 110 validates the integrity of the object of the request and bypasses the AV Scanner 300 by immediately branching to block 126, where the object of the integrity verification request is returned to the requester.

10           FIG. 4 and the following discussion are intended to provide a brief, general description of a suitable computing environment in which certain aspects of the illustrated invention may be implemented. The invention may be described by reference to different high-level program modules and/or low-level hardware contexts. Those skilled in the art will realize that program module references can be interchanged with low-level hardware instructions.

15           Program modules include procedures, functions, programs, components, data structures, and the like, that perform particular tasks or implement particular abstract data types. The modules may be incorporated into single and multi-processor computing systems, as well as hand-held devices and controllable consumer devices (e.g., Personal Digital Assistants (PDAs), cellular telephones, etc.). It is understood that modules may be implemented on a single computing device, or processed over a distributed network environment, where modules can be  
20           located in both local and remote memory storage devices. An exemplary system for implementing the invention includes a general-purpose computer system 500 comprising a bus 501, or other communications hardware and software, for communicating information, and a processor (or processing unit) 505 coupled with bus 501 for processing information. Computer

system 500 further comprises a random access memory (RAM) or other dynamic storage device 502 (referred to as main memory), coupled to bus 501 for storing information and instructions to be executed by processor 505. Computer system 500 also comprises a read only memory (ROM) 503, and/or other static storage device, coupled to bus 501 for storing static information and instructions for processor 505. Mass storage device 504 is coupled to bus 501 for storing information and instructions. In one embodiment, the instructions comprising the file system integrity verification and validation process 100 may be stored on ROM 503. Alternatively, the instructions may be stored within mass storage device 504.

Furthermore, mass storage device 504, such as a magnetic disk or optical disk or other machine-accessible medium, and its corresponding disk drive, can be coupled to computer system 500 for accessing the information and instructions stored thereon via bus 501. Computer system 500 can also be coupled via bus 501 to a display device 521 for displaying information to a computer user such as a network manager. Display device 521 is used to display a graphical user interface to the present invention. Display device 521 can include a frame buffer, specialized graphics rendering devices, a cathode ray tube (CRT), and/or flat panel display. An input device 522, including a device with alphanumeric and other keys, or a voice command input device capable of accepting audio input is typically coupled to a bus 501 for communicating information and command selections to processor 505. Another type of user input device is a cursor control device 523, such as a mouse, a trackball, a pen, a touch screen, or cursor direction keys for communicating direction information and command selections to processor 505, and for controlling cursor movement on display device 521. This input device typically has two degrees of freedom in two axes, a first axis (e.g., the x-axis) and a second axis (e.g., the y-axis), which allows the device to specify positions in a plane. However, this

invention should not be limited to input devices with only two degrees of freedom. Another device that may be coupled to bus 501 is a hard copy device 524 which may be used for printing instructions, data, or other information on a medium such as paper, film, or similar types of media. Additionally, computer system 500 can be coupled to a device for sound recording, and/or playback 525, such as an audio digitizer coupled to a microphone for recording information. Further, the device may include a speaker that is coupled to a digital to analog (D/A) converter for playing back the digitized sounds.

Network interface card 526 is coupled to bus 501. Network interface card 526 is further coupled to an external computer network (not shown). Network interface card 526, in conjunction with appropriate data communications protocols (e.g., the TCP/IP suite of internetworking protocols), provide the means by which an embodiment of the present invention operating on a general-purpose computer system 500 exchanges information with other devices coupled to the same computer network. Modem 527 is coupled to bus 501, and provides an alternate means of exchanging information with other devices for which a modem connection to an external computer network or device (not shown) can be established.

Computer system 500 and the file system integrity verification and validation process 100 stored and executed therein as part of the method and apparatus of the present invention operate in conjunction with an operating system with graphics capability, such as Microsoft's Windows or Windows CE operating systems. Commercially available computer systems implementing the features of general-purpose computer system 500 include a broad range of operating system-based computers, including server computers, desktop computers, workstations, personal digital assistants, devices, or other computer appliances. Furthermore, the present invention may be used in conjunction with various browsers (e.g. Microsoft Internet Explorer or Netscape

Navigator) designed for both conventional and wireless web servers and various electronic mail applications (e.g. Microsoft Outlook, and Lotus Notes) or other messaging applications to yield an operational platform upon which an embodiment of the present invention may be implemented.

5           The storage systems and associated machine-accessible media provide storage of data and executable instructions for the computing device 500. Storage options include hard-drives, floppy-disks, optical storage, magnetic cassettes, tapes, flash memory cards, memory sticks, digital video disks, and the like, and may be connected to the bus 501 by way of an interface. Computing device 500 is expected to operate in a networked environment using logical  
10 connections to one or more remote computing devices through a network interface 526, modem 527, or other communication pathway. Computing devices may be interconnected by way of a network such as a local intranet or the Internet.

          Having described and illustrated the principles of the invention with reference to illustrated embodiments, it will be recognized that the illustrated embodiments can be modified  
15 in arrangement and detail without departing from such principles. For example, while the foregoing description focused on verifying and validating the integrity of an object in the context of on-demand scanning of a disk or collection of files, it will be recognized that the above techniques and analyses can be applied to scanning data in other contexts such as on-access virus scanners for scanning one or more objects, and having comparable limitations.

20           And, even though the foregoing discussion has focused on particular embodiments, it is understood that other configurations are contemplated. In particular, even though expressions such as “in one embodiment” and the like may be used herein, such phrases are meant to generally reference embodiment possibilities, and are not intended to limit the invention to

particular embodiment configurations. As used herein, these phrases may reference the same or different embodiments, and unless expressly indicated otherwise, are combinable into other embodiments. Consequently, in view of the wide variety of permutations to the above-described embodiments, the detailed description is intended to be illustrative only, and should not be taken  
5 as limiting the scope of the invention.

What is claimed as the invention, therefore, is all such modifications as may come within the scope and spirit of the following claims and equivalents thereto.

0  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
1000